PROVISIONAL APPLICATION FOR PATENT COVER SHEET

A/11/60

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

Docket No. 02331-0110P

INVENTOR(s)

LAST NAME	FIRST NAME	M.I.	RESIDENCE (city & either state or foreign country)
CLOPINET	Isabelle	G.	Berkeley, California

TITLE OF THE INVENTION (280 characters max)

APPLICATIONS OF SVMs IN GENOMICS AND CANCER RESEARCH

CORRESPONDENCE ADDRESS

JONES & ASKEW, LLP
2400 Monarch Tower
3424 Peachtree Road, N.E.
Atlanta, Georgia 30326

Attn: James Dean Johnson, Ph.D.

ENCLOSED APPLICATION PARTS (*check all that apply*)

- ☒ Specification — *Number of Pages* 11
- ☐ Drawing(s) — *Number of Sheets* _____
- ☐ Provisional Application Filing Fee
- ☐ Small Entity Statement
- ☐ Other (specify) _____

METHOD OF PAYMENT

- ☐ A check is enclosed to cover the Provisional Application filing fee.
- ☐ The Commissioner is hereby authorized to charge any additional filing fee and credit any refund to Deposit Account No. 10-1215.

FILING FEE: $ _____

The invention was not made by an agency of the U.S. Government nor under a contract with an agency of the U.S. Government.

Respectfully submitted,

SIGNATURE: James Dean Johnson Date: October 27, 1999

TYPED OR PRINTED NAME: James Dean Johnson, Ph.D. Reg. No. 31,771

☐ Additional inventors are being named on separately numbered sheets attached hereto.

"Express Mail" Mailing Label Number EL397836790US

Applications of SVMs in Genomics and Cancer Research

September 15 to October 27, 1999
Isabelle Guyon
Clopinet
955 Creston Road, Berkeley, CA 94708
(510) 524-6211, isabelle@clopinet.com

I. Introduction

This report presents the results of preliminary experiments that we did on the **Leukemia data of Golub et al.** Our goal is to device a method for selecting the best SVM for the task, using only training data. This involves various "model selection" criteria, including the leave-one-out error rate and the size of the margin, rescaled by the largest distance between patterns. We reserved the test set for the "final test" and did not touch it yet.
We also present a **parallel algorithm** for SVM that was invented by Ross Baldick.
During this report period, we did also various explorations on the Brown et al data, compared SVM training algorithms, and wrote a white paper on SVM applications. These other tasks are or will be described in separate documents.

II. Paper review and description of the tasks

In their paper:
Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Golub et al, Science Vol 286, Oct 1999.
the authors present methods for analysing gene expression data obtained from DNA micro-arrays in order to classify types of cancer.

Data set:
Their method is illustrated on Leukemia data. The problem is the distinction between two variants of Leukemia (ALL and AML).
Their training set consists of 38 samples (27 ALL and 11 AML). Their test set has 34 samples (20 ALL and 14 AML) collected under different experimental conditions. All samples have 7129 attributes (or features) corresponding to some normalized gene expression value extracted from the micro-array image.

Tasks:
The authors investigate two tasks:
- **Class prediction** (supervised learning): after training a classifier on the training set, including the ALL/AML labeling information, they try to predict the ALL/AML class labels on the test set.
- **Class discovery** (unsupervised learning): They remove the class labels and use a clustering technique to find whether the distinction ALL/AML can be discovered automatically.

In this report, we concentrate on the first task: class prediction.

Exhibit 2

The authors also address an important sub-problem: that of attribute (or feature) selection. In this case, **gene selection**. The device a method that selects 50 genes out of 7129.

Algorithms:

Gene selection:

To reduce the dimensionality of input space, the authors use the following technique: find the features (genes) that resemble most the target vector (or its opposite), using the following metric:

$$P = (m_1 - m_2) / (s_1 + s_2)$$

where m_1 and s_1 are the mean and standard deviation values of the given feature on class 1 examples (e.g. ALL examples). Similarly m_2 and s_2 are the mean and standard deviation values of the given feature on class 2 examples (e.g. AML examples).

They reduce the space from 7129 to 50 genes. In a more detailed technical memorandum, they mention that a number of genes from 3 to 200, selected with this method, give similar results. This suggested to us that perhaps only 2 genes selected in a better way would suffice.

Class prediction:

The authors use a **linear classifier**, with the following decision function:

$$D(\mathbf{x}) = \mathbf{w}.(\mathbf{x}-\mathbf{b})$$

where $\mathbf{x}$ is an input vector (the gene expression of a patient) and $\mathbf{w}$ and $\mathbf{b}$ are a weight and a bias vector computed from the training data as follows:

$$w_i = (m_1 - m_2) / (s_1 + s_2)$$

where m_1 and s_1 are the mean and standard deviation values of the feature (gene expression) number i on class 1 examples. Similarly m_2 and s_2 for class 2.

$$b_i = (m_1 + m_2)/2$$

This classification method bears similarity with Bayesian classifiers assuming Normal data distribution, as explained in a detailled TM available from the authors web site.

Class discovery:

The authors use Self Organizing feature Maps (SOM), a well know neural network technique invented by Kohonen.

Normalization:

All features are normalized by subtracting the mean feature value on the training examples and dividing by the standard deviation, also computed on the training examples.

Methodology:

Leave-one-out

To select between algorithm variants, the leave-one-out method is use: one example of the training set is taken out. Training is performed on the remaining examples. The left out example is used to test. The procedure is iterated over all examples.

Feature selection/normalization

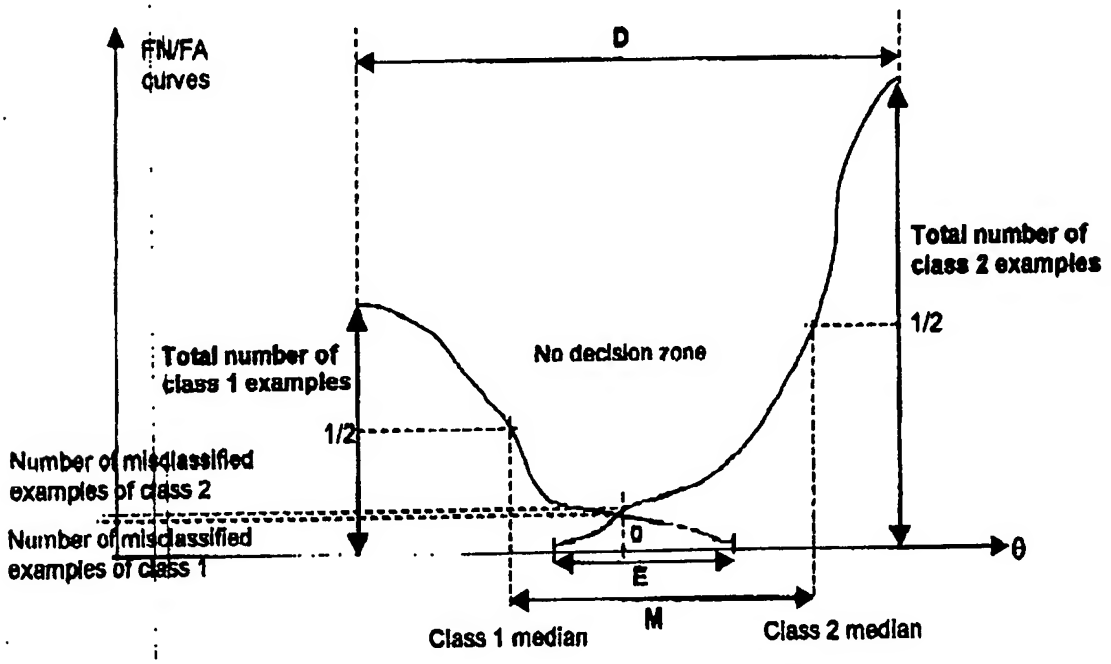
- Feature (gene) selection, training and parameter tuning is performed on the training set only.
- The features selected using the knowledge of the class labels cannot be used for unsupervised learning.
- Feature selection performed on all training examples, therefore this biases a little bit the leave-one-out prediction.
- Normalization is performed using the mean and standard deviation computed on the training examples only. When applied to the leave-one-out, these quantities are recomputed for every left out pattern.

Rejection

In some cases, it is better to refuse making a decision rather than making a wrong decision (e.g. that could potentially lead to the wrong treatment). The classifier score is used to assess classification confidence. Below a certain confidence value, the example is "rejected", that is no classification decision is taken.

III. Methodology improvements

Our exploratory experiments indicated that the problem is simple and that there are very few classification error. In particular, it is possible to find classifiers with zero leave-one-out error. In order to be able to compare classifiers, we plot false negative/false positive curves as described below.



Red: Number of examples of class 1 whose decision function value is smaller than or equal to θ.
Blue: Number of examples of class 2 whose decision function value is larger than or equal to θ.

FN/FP curves

The problem we are interested in is a two-class problem. The class labels are (-1) for class 1, the "negative class" and (+1) for class 2, the "positive class".
The classifiers we are interested in (Golub et al or SVM) make their decision according to the value of a decision function D(x) of an input vector x, e.g.:

If $D(x)<\theta$, classify x in class 1
If $D(x)>\theta$, classify x in class 2

D(x) already incorporates a bias, which is a parameter optimized by training on the training set. Threshold θ is another parameter which is determined by cross-validation. It is used in some applications for which classifying an example of class 1 into class 2 (type I errors) is more severe than the opposite (type II errors). By adjusting θ, one can monitor the ratio of one type of error over the other.
One way of reading the fn/fp curves is to think of the red curve as the number of false positive (type I errors) as a function of θ and the blue curve as the number of false negative (type II errors) as a function of θ. By choosing θ one can monitor the tradeoff between type I and type II errors.
The fn/fp curves are obtained by calculating D(x) by cross-validation using the leave-one-out method. After θ and other parameters are adjusted, new fn/fp curves can be obtained from the test data.

Classifier quality

In this application, we fix $\theta=0$. We use the fn/fp curves for the purpose of evaluating classifier quality.
The red curve is the number of examples of class 1 whose decision function value is smaller than or equal to θ. The blue curve is the number of examples of class 2 whose decision function value is larger than or equal to θ. We derive from these curves 3 parameters that characterize classifier quality:

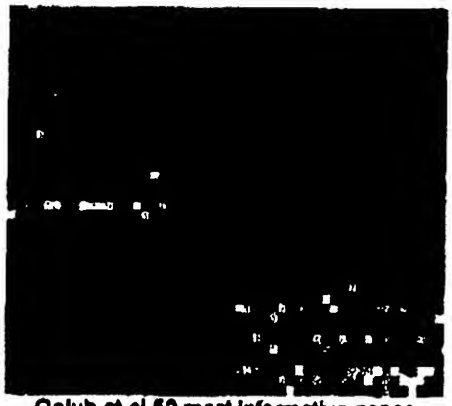
- The **total number of classification errors** (sum of type I and type II errors at $\theta=0$). The fewer errors, the better the classifier.
- The **extremal margin E/D** (the re-scaled difference between min(D(x)) for class 2 examples and max(D(x)) for class 1 examples). It indicates the performance on the worst classified patterns. The extremal margin is negative if there are classification error. If there are no classification errors, it is positive. In some cases it may be positive and yet there are some remaining errors (the decision threshold in not in the margin area).The larger the extremal margin is, the better the classifier.
- The **median margin M/D** (the re-scaled difference between median(D(x)) for class 2 examples and median(D(x)) for class 1 examples). The median margin is usually positive. It indicates how well the two classes are separated on average. The larger the median margin is, the better the classifier.

Rejection

One can define a zone in which no decision of classification is made on either side of the decision threshold (e.g. light yellow region). Within this zone, classification decisions are considered uncertain.

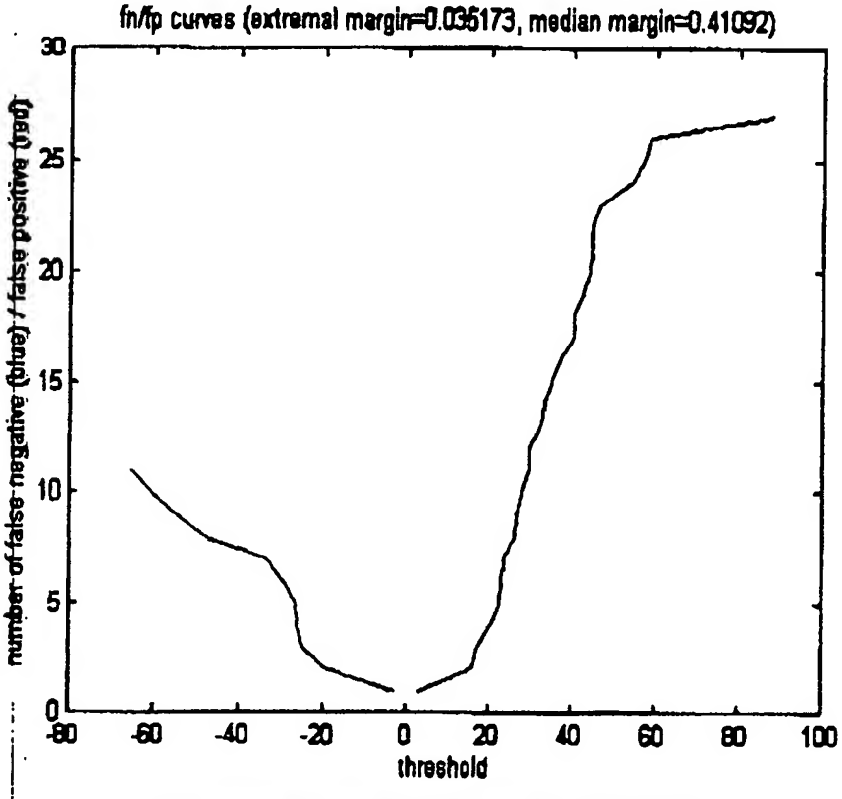
IV. Reproducing the baseline results

We reproduced the baseline results of the Golub et al paper. We extracted from their paper the 50 informative genes that they selected:



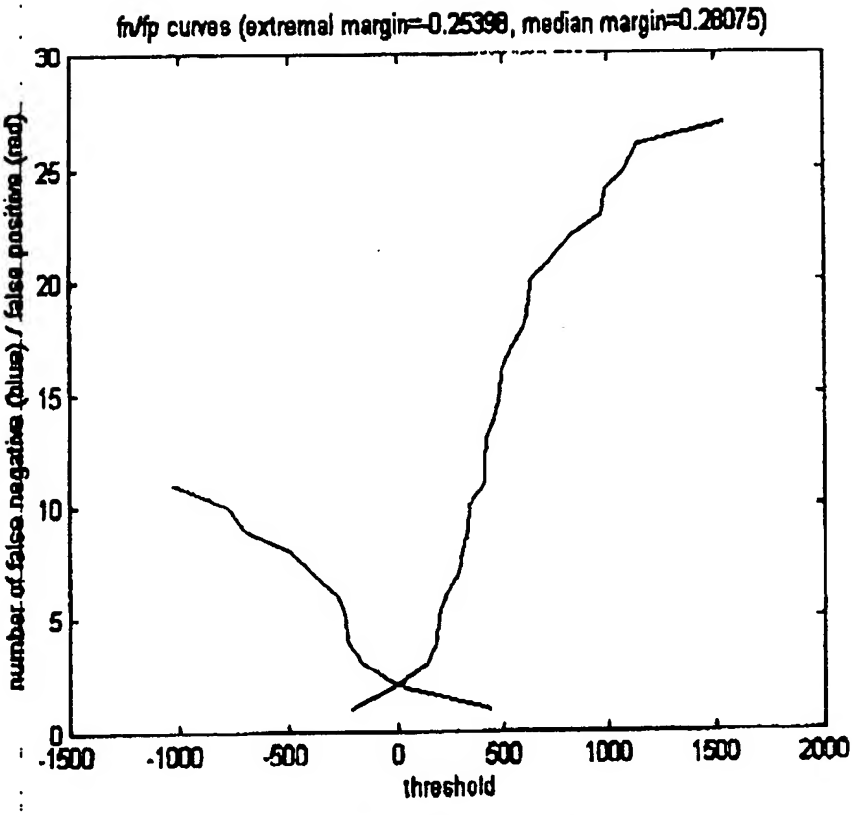
Golub et al 50 most informative genes
for 38 training examples (27 ALL top, 11 AML bottom).

We implemented their classification technique and obtained the following fn/fp curves by cross-validation (leave-one-out):



Golub et al best results on their selection of 50 genes.

This confirms that with the leave-one-out method they have zero error. Their extremal margin is small: the examples that are hardest to classify are not classified with high confidence.



Golub et al without gene selection

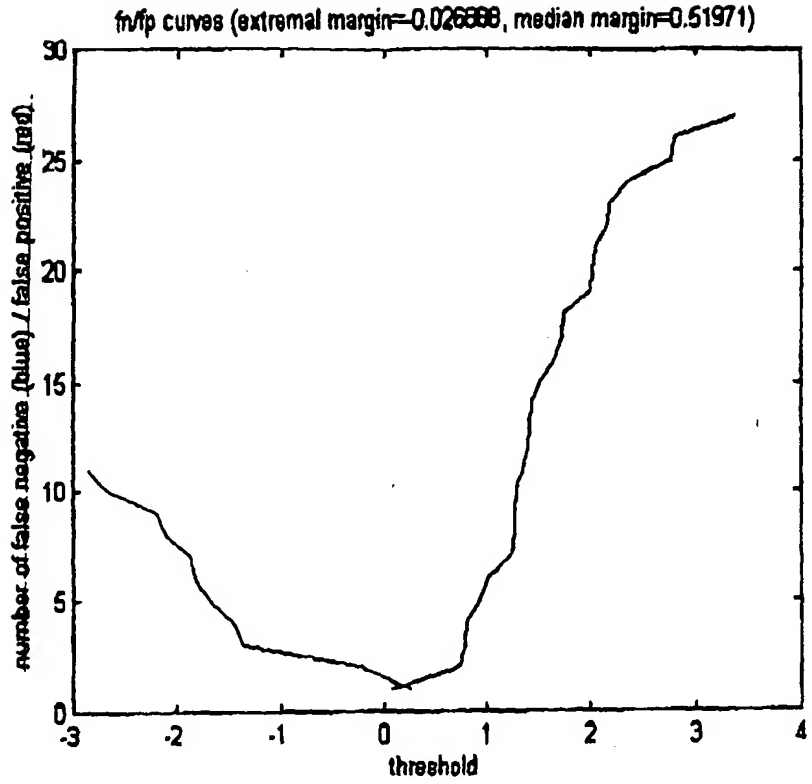
We also tried their classification technique on the while set of 7129 genes (no gene selection). This test reveals the classification method of Golub et al performs poorly without first reducing the dimensionality of input space. The leave-one-out error rate is 4/38. The extremal margin is very negative.

SVM results

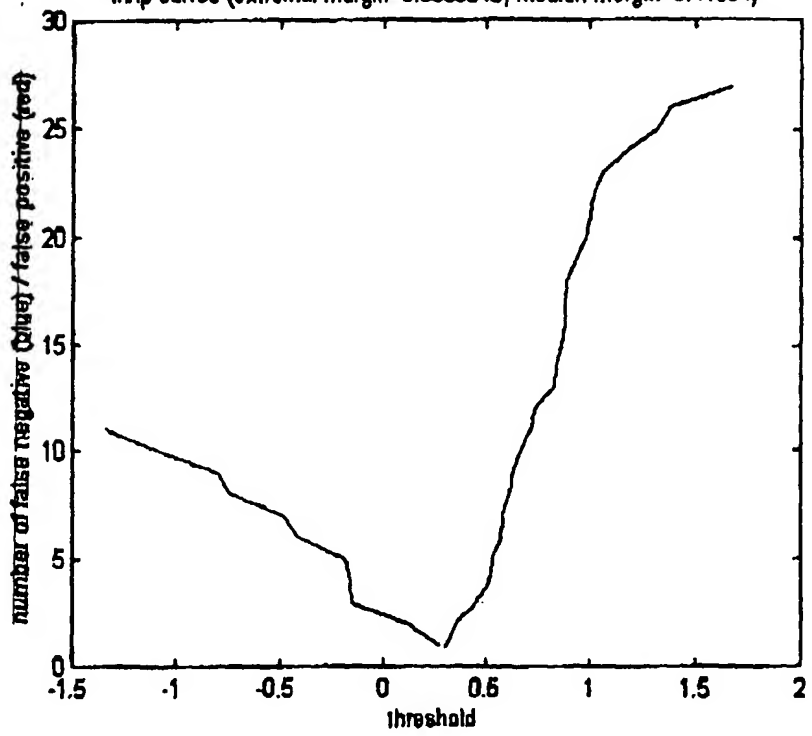
Our exploratory experiments indicated that the training set is linearly separable. We tried linear SVM, polynomial SVM and radial SVM. Linear SVMs outperform other methods. We concentrated our efforts on linear SVM and on the improvement of the feature (gene) selection method.

We trained an SVM classifier on their 50 features (genes). In this experiment (figure on next page), it is arguable which classifier is best: SVM have 1 error and a small negative extremal margin, but the SVM median margin is larger than the classifier of Golub et al trained on the same features.

SVMs can also be trained without feature (gene) selection. The linear SVM obtained (figure on next page) has only 2 errors and a small positive extremal margin. Its median margin is much larger than Golub et al on all genes. Overall, it is clearly a superior classifier to the baseline classifier trained in the same conditions.



Linear SVM trained on the 50 best genes of Golub



Linear SVM without gene selection

SVM-based feature selection

The feature selection method of Golub et al is designed to work best with their classification technique. Moreover, it is rather crude. We designed and implemented an SVM-based feature selection technique which proved to be superior and allow us to narrow down the number of genes to only two.

Feature selection algorithm description:

Method 1 (combinatorial and slow):

SVMs are trained using subsets of input features of the same size. The resulting classifiers are ranked in order of classifier quality (as measured, for instance by the ratio of margin size over the largest distance between patterns). The subset of features yielding the best classifier is selected.

This method is not practical if the number of features is large and/or the size of the subsets is large, because of computational considerations. Therefore, we complement is with a second method, weaker but faster.

Method 2 (sub-optimal but fast):

A first SVM classifier is trained on all the features. The features are then ranked in order of increasing weight:

$$w_i = \text{sum}\{ \alpha_k y_k x_{ki} \}$$

where the sum runs over the support vectors x_k of class polarity y_k (+1 or –1) and Lagrange multiplier α_k. For linear SVMs, these weights are the weights of the linear classifier itself, in feature space. For non-linear SVMs, these weight are proportional to an average of all the weights that involve input feature i.

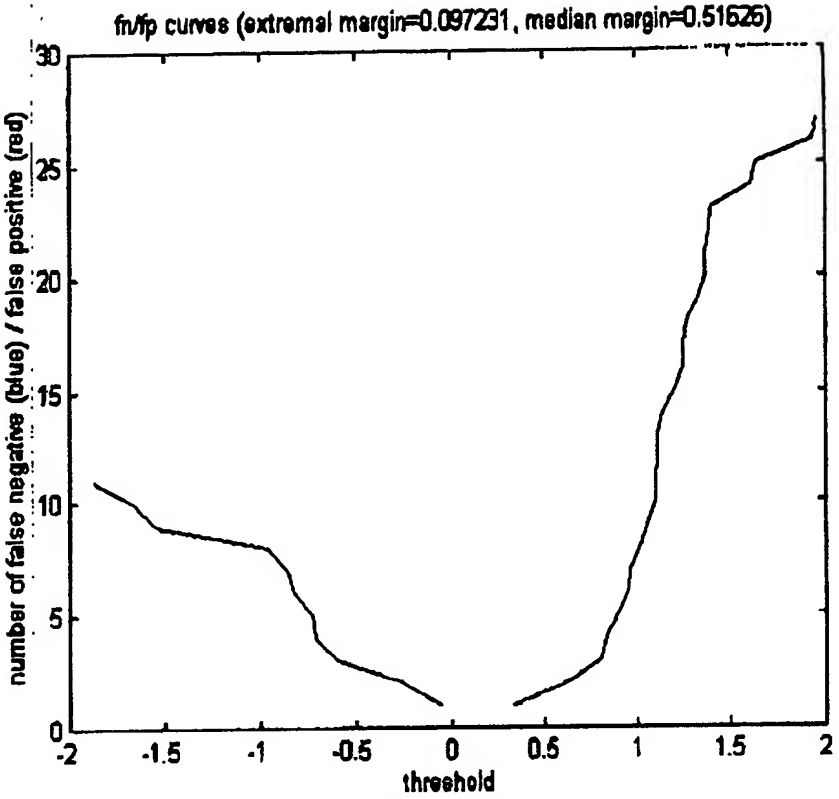
The input features corresponding to the largest weights are kept. This method is justified by the fact that removing the features with smallest weight least perturbates the solution.

After feature pruning, the classifier needs to be retrained. The procedure may be iterated.

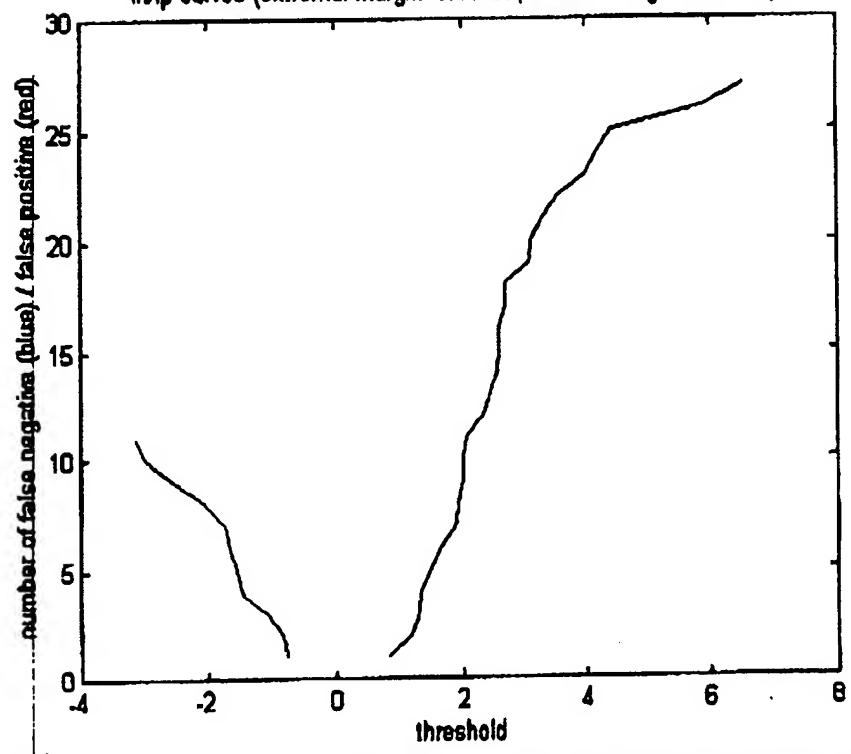
Combined method:

Method 2 is used to downsize feature space as much as possible before using method 1.

In one of our implementations, we first removed enough features to reach a number of input features that is a power of 2. We then iterated the feature pruning method by dividing by two the number of features at each step. For each classifier that we trained, we measured classifier quality with the ratio of margin size over the largest distance between patterns. Quality as a function of number of input features was plotted. This curve exhibits a maximum. We selected the feature subset of maximum quality. On that set, we used method 1 to reach a subset of only 2 features.



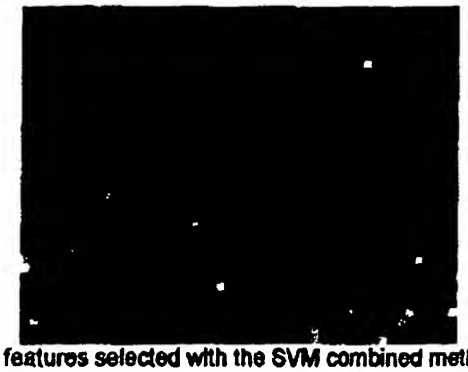
SVM trained on 50 genes selected with SVM pruning method 2.



SVM trained only on two genes (M23197 and M81933) selected with the combined method.

Feature selection experiments:

In order to compare with the baseline recognizer, we used feature pruning method 2 to downsize feature space to 50 features (figure below). It is interesting to notice that these features do not look as orderly as the features selected by Golub et al's method and yet perform better. This is not so surprising: all of Golub et al features are very correlated and therefore carry less information than ours. The SVM classifier that we trained on these features is better than the baseline classifier in all respects: larger extremal and median margins (figure on previous page).



50 features selected with the SVM combined method

We used the combined method to select only two genes to perform the separation (figure on previous page). We also obtained perfect separation by cross-validation. The extremal and median margins remain comfortable. The baseline method of Golub et al could never achieve such a result.

The two genes selected are:
M81933-at : CDC25A Cell division cycle 25A
M23197-at ; CD33 CD33 antigen (differentiation antigen)

Gene M81933 and M23197

In two dimensions, it is possible to visualize the solution obtained.

SVM two-dimensional separation with genes M81933 and M23197

V. Parallel algorithm for SVM

We briefly describe the algorithm of Ross Baldick et al for parallel implementation of SVMs:

Assume N parallel processors.

Initialization: Divide the training set into N subsets of equivalent size. Each processor is assigned a subset (initial working set).

Step 1: Each processor trains an SVM with its working set.

Step 2: The support vectors found are broadcasted and all the support vectors found by all processors are added to the working sets of each processor.

Step 1 and 2 are iterated until convergence.

Properties:

- Convergence is garanteed.
- The algorithm was invented for classification, but works also for regression.

VI. Further work

Next, we will refine our feature selection and model selection method and clean up our code. While we already outperform Golub et al comparing cross-validation results, we still want to build more confidence in our classifier to be sure that we will outperform them on the test set too. This process will include refining our classifier quality measurements.

We will also solve the class discovery problem (unsupervised learning), using SVM.